複合誤り訂正符号について (符号と暗号の代数的数理)

AUTHOR(S):

藤沢, 匡哉; 前田, 秀介; 阪田, 省二郎

CITATION:

藤沢, 匡哉 ...[et al]. 複合誤り訂正符号について (符号と暗号の代数的数理). 数理解析研究所講究録 2004, 1361: 162-170

ISSUE DATE:

2004-04

URL:

http://hdl.handle.net/2433/25269

数理解析研究所講究録 1361 巻 2004 年 162-170

複合誤り訂正符号について

藤沢 匡哉
Masaya Fujisawa
東京理科大学 工学部第二部 経営工学科
Department of Industrial Manegement and Engineering, Tokyo University of Sience
e-mail: fujisawa@ms.kagu.tus.ac.jp

前田 秀介
Shusuke Maeda
電気通信大学 情報工学専攻
Course in Computer Science and Information Mathematics, The University of Electro-Communications
e-mail: maeda@ice.uec.ac.jp

阪田 省二郎
Shojiro Sakata
電気通信大学 情報通信工学科
Department of Information and Communication Engineering, The University of Electro-Communications
e-mail: sakata@ice.uec.ac.jp

平成 15 年 11 月 7 日

1 はじめに

実際の通信路には, ランダム誤りとバースト誤りが混在して発生しているとみなせるものが多い. このようなランダム誤り (長さ 1 のバースト誤り) を含む様々な長さをもつバースト誤りの任意の組合せを複合誤りという. ランダム誤り訂正におけるハミング距離 (重み) に代わって様々なバーストの組合わせを同時に取り扱うための尺度として導入された Gabidulin[1] の組合せ距離の一般化である複合距離 (重み) [2], [3], [4] に基づいて, 複合誤り訂正符号が導入されている.

これらの符号の性能評価を行うためには複合重みを効率的に計算する必要がある. そこで, まず, トレリスに基づいて任意の語に対する複合重みの計算法を与える. 次に, ハミング重みに関して和田山等 [5] が提案したコセット追加法を複合重みに関して拡張することを考える. つまり, シンドロームトレリスと重み計算トレリスの直積である 3 次元トレリスにより, コセット代表元とその重みを計算するアルゴリズムを提案する. このアルゴリズムを用いてコセット追加を行うことにより複合誤り訂正能力は変えずに符号化率を上げることができる. また, この方法の適用結果として得られた良い複合誤り訂正符号のいくつかの例を示す.

最後に, 複合誤り訂正符号の復号法について述べる. これまでに, これらの符号に対する復号法として順序型限界距離復号法 [3] や Step-by-step 復号法 [4] が与えられていた. 前者は, 代数的な複合誤り訂正符号の特別なクラスについて代数的な復号法を与えている. 後者は, コセット代表元とその重みの表を利用し, 辞書順にバースト誤りを段階的に取り除く方法であり, その計算量は指数的である. 本研究では, コセット代表元計算アルゴリズムにおける 3 次元トレリス上の重み計算を距離計算となるように辺と重みを修正することにより, 任意の複合誤り訂正符号に対する復号法を提案する.

以下に, 本論分の構成を示す. 2 節では, 準備として複合誤り訂正符号に対する基本的な概念と定義を与える. 3 節ではトレリスに基づいた複合重みの計算アルゴリズムを与え, 4 節では複合誤り訂正符号に対する符号追加アルゴリズムを提案し, アルゴリズムに

より得られた良い符号の例を示す. 最後に, 任意の複合誤り訂正符号に対する復号法を提案する.

2 準備

複合距離・重みの定義は [4] とほぼ同じものを取り扱う. 以下に概略を示す. 有限体 F_q 上の線形符号 C を考え, 符号長 n に対して, $N := \{1, 2, \cdots, n\}$ とする.

バースト長の集合 $B := \{b_1, \cdots, b_m\} \subset N$ とバースト重みの集合 $\Pi := \{\pi_1, \cdots, \pi_m\}$ の組を仮定する. ここで, 自然数 $b_1, \cdots, b_m$ と正実数 $\pi_1, \cdots, \pi_m$ は以下の条件を満たす.

$$\begin{array}{ccccccccc} b_1 = 1 & < & b_2 & < & \cdots & < & b_m, \\ \pi_1 = 1 & < & \pi_2 & < & \cdots & < & \pi_m, \\ \frac{\pi_1}{b_1}(=1) & > & \frac{\pi_2}{b_2} & > & \cdots & > & \frac{\pi_m}{b_m}. \end{array}$$

さらに, $b_0 := 0$, $\pi_0 := 0$ とする.

任意のベクトル $\underline{v} = (v_i)_{1 \le i \le n} \in F_q^n$ に対し, バーストの始点と終点は, $\mathrm{in}(\underline{v}) := \min\{i \in N \mid v_i \neq 0\}$, $\mathrm{fn}(\underline{v}) := \max\{i \in N \mid v_i \neq 0\}$, そのバースト長は $\mathrm{bl}(\underline{v}) := \mathrm{fn}(\underline{v}) - \mathrm{in}(\underline{v}) + 1$ と定義される.

- **b_i-バースト** (burst) $\underline{d}$:
  バースト長が $b_{i-1} < \mathrm{bl}(\underline{d}) \le b_i$ であり, 重みが $\mathrm{w}(\underline{d}) := \pi_i$, $1 \le i \le m$ を満たす語 $\underline{d} := (d_i)_{1 \le i \le n} \in F_q^n$.
  (単に, b_i-バーストをバーストと呼ぶ.)

- **b_i-バースト $\underline{d}$ の被覆** :
  集合 $\mathrm{supp}(\underline{d}) := \{j \in N \mid \mathrm{in}(\underline{d}) \le j \le l\}$,

$$l := \begin{cases} \mathrm{in}(\underline{d}) + b_i - 1 & \text{if } \mathrm{in}(\underline{d}) < n - b_i + 1, \\ n & \text{otherwise.} \end{cases}$$

- **語 $\underline{v}$ の被覆** (cover) :
  $\underline{v} = \sum_{i=1}^{s} \underline{d}_i$ かつ $\mathrm{supp}(\underline{d}_i) \cap \mathrm{supp}(\underline{d}_j) = \emptyset$, $i \neq j$ を満たすバースト $\underline{d}_1, \cdots, \underline{d}_s$ の集合 $c = \{\underline{d}_1, \cdots, \underline{d}_s\}$. $\underline{v}$ に対して取り得るすべての被覆の集合を $\mathrm{Cov}_{\underline{v}}$ と記述する.

- **語 $\underline{v}$ の被覆 $c := \{\underline{d}_1, \cdots, \underline{d}_s\}$ の重み**:
  $\mathrm{w}(c) := \sum_{i=1}^{s} \mathrm{w}(\underline{d}_i)$;

- **$\underline{v}$ の複合重み** (単に**重み**):
  $\mathrm{w}(\underline{v}) := \min\{\mathrm{w}(c) \mid c \in \mathrm{Cov}_{\underline{v}}\}$;

- **符号 C の最小 (複合) 重み**:
  $\mathrm{w}(C) := \min\{\mathrm{w}(\underline{u}) \mid \underline{u} \in C, \underline{u} \neq 0\}$;

- **2 つの語 $\underline{u}, \underline{v} \in F_q^n$ の複合距離** (単に**距離**):
  $\mathrm{d}(\underline{u}, \underline{v}) := \mathrm{w}(\underline{u} - \underline{v})$;

- **符号 C の最小 (複合) 距離**:
  $\mathrm{d}(C) := \min\{\mathrm{d}(\underline{u}, \underline{v}) \mid \underline{u}, \underline{v} \in C, \underline{u} \neq \underline{v}\}$;

任意の線形符号の最小距離と最小重みは等しく, 最小距離 $\delta := \mathrm{d}(C)$ をもつ符号 C は $\frac{\delta}{2}$ より小さい重みを持つ任意の複合誤りを訂正できる [2], [3]. 最小距離 δ は B, Π に依存するため, 符号の複合誤り訂正能力, 特に, 最小複合距離 δ は, B と Π に依存する. これらが固定された上で, 訂正可能な誤りパターンは最小複合距離 δ によって一意に決まる.

例 1 $B = \{1, 3, 7\}$, $\Pi = \{1.0, 1.8, 3.5\}$, $\delta = 10.0$ ならば, 重みが 5.0 より小さな複合誤りパターンの中で極大なものは, $3 \times 1b + 1 \times 3b$ (複合重み 4.8), $1 \times 1b + 1 \times 7b$ (複合重み 4.5) の 2 種類となり, $2 \times 1b + 1 \times 2b$ や $1 \times 1b + 1 \times 5b$ などのより小さな重みをもつ残りのすべてのパターンは訂正可能である. ここで, $3 \times 1b + 1 \times 3b$ は, 3 個のランダム誤り (長さ 1 のバースト誤り) と 1 個の長さ 3 のバースト誤りを表している.

最終的な符号の複合誤り訂正能力は (B と Π を前提とした上で) 最小複合距離によって定まる. 同じ符号でも, B と Π の設定により訂正能力が異なることがあり, 本論文では, B と Π の値は試みのものに設定し, それを前提として符号毎に定まる最小複合距離を算出している. 実際には, 与えられた通信路に最も適合する B と Π の値を採用すべきであるが, それを適切に決定する問題は, 符号の選択とも絡んで, 難しい面を含んでおり, 本論分では取り扱わないことにする.

本論分の残りの部分では, 2 元線形符号 C, すなわち, 2 元体 $F_2 = \{0, 1\}$ 上での F_2^n の線形部分空間を考えることにする. k, d, δ を次元, 最小ハミング距離, 最小複合距離とする. このとき, C を $[n, k, d, \delta]$ あるいは, より詳細に $[n, k, d, \delta]_{B,\Pi}$ と表記する.

3 重み計算

複合誤り訂正符号を取り扱う上で, 任意の符号語の高速重み計算は必要不可欠である.

Problem: (重み計算)

Given $\underline{v} = (v_1, \cdots, v_n) \in F_2^n$;
Find $\mathrm{w}(\underline{v}) := \min_{c \in \mathrm{Cov}_{\underline{v}}} \sum_{\underline{d}_j \in c} \mathrm{w}(\underline{d}_j)$;

この問題に対して, トレリスに基づいた任意の語 $\underline{v} \in F_2^n$ に対する重み計算アルゴリズムを提案する.

各時刻 t, $1 \le t \le n$ において, $M = (\sum_{i=1}^{m} b_i) + 1$ 個の状態 S_{ij}, $j = 1, \cdots, b_i$, $i = 0, \cdots, m$ を考える. ここで, 時刻 t は語 $\underline{v} = (v_1, \cdots, v_n)$ の t-番目の元 v_t と対応している. 各 t, $1 \le t \le n$ に対して, 状態 S_{00} と状態 S_{ij}, $j = 1, \cdots, b_i$, $i = 1, \cdots, m$ はバースト誤りなし, b_i-バーストの j-番目の位置を表している. このトレリスは各時刻 t, $1 \le t \le n$ において, 各状態に対応する M 個の頂点 V_{ij}^t をもつ. ただし, 時刻 $t = 0$ においては状態 S_{00} に対応する 1 つの頂点 V_{00}^0 のみである. さらに, これらの頂点に付け加えて時刻 $t-1$ の頂点から時刻 t の頂点を結ぶ以下のような辺をもつ.

(i) $j = b_i$, $0 \le i \le m$
  (a) (V_{ij}^{t-1}, V_{00}^t) if $v_t = 0$,
  (b) (V_{ij}^{t-1}, V_{k1}^t), $1 \le k \le m$ if $v_t \neq 0$,
(ii) $1 \le j < b_i$, $1 \le i \le m$
  $(V_{ij}^{t-1}, V_{i,j+1}^t)$.

(i-a) の辺はバースト外にあるといい, (i-b), (ii) の辺は b_i-バースト内にあるという. バーストではない辺の重みは 0 であり, b_i-バースト内の辺の重みは π_i/b_i である.

次に, 任意の被覆の重みを計算しやすくするために, 時刻 $n + b_m$ までトレリスを拡張する. ここで, $n < t \le n + b_m$ に対して, $v_t = 0$ とし, 対応する辺は以下の通りとなる:

(i) (V_{ij}^{t-1}, V_{00}^t), $j = b_i$, $0 \le i \le m$
(ii) $(V_{ij}^{t-1}, V_{i,j+1}^t)$, $1 \le j < b_i$, $1 \le i \le m$.

$t = n + b_m$ においては, 時刻 $t = 0$ と同様に 1 つの頂点 $V_{00}^{n+b_m}$ となる.

与えられた語 $\underline{v}$ に対して, 以下のアルゴリズムは t に関して再帰的に時刻 0 における状態 S_{00} から 時刻 t における状態 S_{ij} までの取り得るすべてのパスに含まれる辺の重み総和の最小値である $w(t,i,j)$ を計算する. 最終的に, 語 $\underline{v}$ の重み $\mathrm{w}(\underline{v})$ が得られる.

アルゴリズム 1 (重み計算)

Input $\underline{v} = (v_i)_{1 \le i \le n}$.

Output $\mathrm{w}(\underline{v})$ $(= w(n + b_m, 0, 0))$.

Step 1 Set $w(0,0,0) := 0$, $w(t,i,j) := \infty$ for $1 \le j \le b_i$, $1 \le i \le m$, $0 \le t \le n + b_m$; $v_i := 0$ for $n < i \le n + b_m$; $t := 1$;

Step 2 Calculate

$$w(t,i,j) := \begin{cases} \min\limits_{0 \le k \le m} \{w(t-1,k,b_k)\} & \text{if}(v_t = 0, i = 0, j = 0), \\ \min\limits_{0 \le k \le m} \{w(t-1,k,b_k)\} + \pi_i/b_i & \text{if}(v_t \neq 0, i \neq 0, j = 1), \\ w(t-1,i,j-1) + \pi_i/b_i & \text{if}(i \neq 0, j \neq 1). \end{cases}$$

Step 3 If $t = n + b_m$, then stop else $t := t + 1$ and go to Step 2.

各時刻における状態数は M であり, 各状態に対して高々 m 回の加算, 比較が必要となるので, 上記のアルゴリズムの計算量は $O(nmM)$ となる.

例 2 バースト長集合を $B = \{1, 3\}$, バースト重み集合を $\Pi = \{1, 1.5\}$ と仮定する. 語 $\underline{v} = (1,0,1,1,0,1,1) \in F_2^7$ に対して, アルゴリズムを適用する. 図 1 には, これらの条件に対応するトレリスが示されており, 図 2 には, 重み $\mathrm{w}(\underline{v})$ をもつ最小パスが他の生き残りパスと共に示されている. アルゴリズム 1 により, 語 $\underline{v}$ の重みは 4.0 と分かる. 図 1, 2 において, 点線は 0, 実線は 1, 鎖線は 0, 1 を表している.

例 3 B と Π の様々な選択に対して, 我々は以下に示す生成多項式 [7] により定義される短縮巡回符号 C の最小距離 $\delta = \mathrm{d}(C)$ を計算し, 各符号の性能を調査

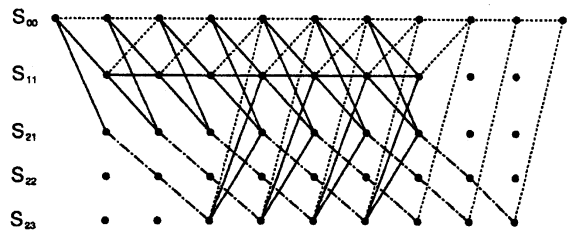

図 1: $\underline{v}$ の重み計算トレリス

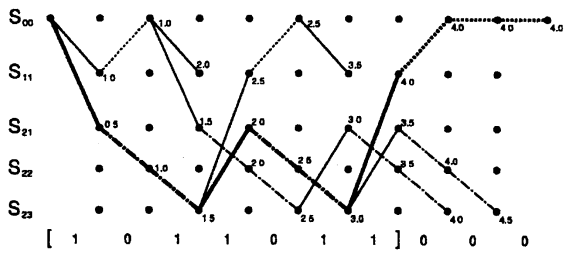

図 2: w($\underline{v}$) に対応する生き残りパス

する.

$$g_1 = x^{13}+x^{11}+x^8+x^7+x^6+x^3+1,$$
$$g_2 = x^{13}+x^{12}+x^{11}+x^{10}+x^7+x^6+x^5+x^4+x^2+1,$$
$$g_3 = x^{14}+x^{12}+x^{11}+x^9+x^8+x^7+x^6+x^4+x^2+1,$$
$$g_4 = x^{16}+x^{13}+x^{11}+x^8+x^6+x^4+x^3+1,$$
$$g_5 = x^{17}+x^{16}+x^{13}+x^{12}+x^{10}+x^8+x^7+x^5+x^2+1,$$
$$g_6 = x^{17}+x^{13}+x^{10}+x^7+x^3+x^2+x+1,$$
$$g_7 = x^{18}+x^{17}+x^{13}+x^{12}+x^{11}+x^{10}+x^9+x^8+x^5+1.$$

各 B と Π の組に対して, 符号長 $n=27$ で生成多項式 g をもつこれらの符号の最小距離 δ と追加複合誤り訂正パターンを表 3 に示す. ここで, 追加複合誤り訂正パターンとは普通のランダム訂正以外に複合誤りパターンとして訂正できる誤りのパターンを意味する. 例えば, $(1\times 5b, 1\times 1b+1\times 2b)$ はこの符号が 1 個の 5-バースト誤り, あるいは, 1 個の 1-バースト誤り (ランダム誤り) と 1 個の 2-バースト誤りの組合せがさらに訂正できることを示している.

例 4 (列置換) $B := \{1,3,7\}$, $\Pi := \{1.0, 1.8, 3.5\}$ とする. 以下の生成行列 G により定義される [27, 9, 10] 準巡回符号 C [8] は最小距離 $\delta = 7.1$ をもつ.

$$G = \begin{bmatrix}
100000000100010010111001011\\
010000000010001001111100101\\
001000000101000100111110010\\
000100000010100010011111001\\
000010000001010001101111100\\
000001000100101000010111110\\
000000100010010100001011111\\
000000010001001010100101111\\
000000001000100101110010111
\end{bmatrix}.$$

G の列置換によって得られる生成行列 G' により定義される線形符号 C' は最小距離 $\delta = 7.3$ をもつ (最小ハミング距離は変化せず, $d=10$ である). 従って, 符号 C' の訂正可能な誤りパターンは $3\times 1b$, $2\times 3b$, $1\times 7b$ となる.

$$G' = \begin{bmatrix}
110000000100001000101101101\\
000000101000100000111110110\\
000000001111000110001101100\\
010000011000111100101100000\\
001000101001001100010100101\\
000010111100001100001001010\\
000001100000101110100101001\\
010000101001001001100001110\\
000100110000000110111001100
\end{bmatrix}.$$

表 1: 各巡回符号の複合誤り訂正能力 (符号長 $n=27$, d: 最小ハミング距離)

k	d	g	B	Π	δ	Extra
14	5	g_1	{1,5}	{1.0,2.0}	5.0	$1\times 5b$
14	6	g_2	{1,6}	{1.0,2.9}	5.9	$1\times 6b$
13	6	g_3	{1,6}	{1.0,2.9}	5.9	$1\times 6b$
13	6	g_3	{1,2,5}	{1.0,1.9,2.9}	5.9	$1\times 5b$, $1\times 1b+1\times 2b$
11	6	g_4	{1,7}	{1.0,2.9}	5.9	$1\times 7b$
10	8	g_5	{1,7}	{1.0,3.9}	7.9	$1\times 7b$
10	8	g_5	{1,2}	{1.0,1.9}	7.8	$2\times 2b$
10	8	g_6	{1,7}	{1.0,3.9}	7.9	$1\times 7b$
10	8	g_6	{1,2}	{1.0,1.9}	7.8	$2\times 2b$
9	8	g_7	{1,8}	{1.0,3.9}	7.9	$1\times 8b$

このように, 生成行列の列置換は符号の最小距離を増すことができる. G の列置換は数多く存在するので, 最も大きな最小距離 δ を与える最良な置換をみつけることは一般に困難である.

次の節では, 与えられた符号に対してコセット追加によって良い符号を構成する方法について考える.

4 コセット追加と最小距離の計算

この節では, コセット追加により与えられた符号から複合誤り訂正能力は変えずにより高い符号化率もつ符号を構成する方法について議論する. この方法はランダム誤り訂正符号に対する方法 [5] を拡張したものである. C は検査行列 $H = [\underline{h}_1, \cdots, \underline{h}_n]$ によって定義される最小距離 δ をもつ複合誤り訂正符号とする. ここで, C は単に $[n,k,\delta]_{B,\Pi}$ と表記する. 任意の $\underline{v} \in F_2^n$ に対して, C のコセットは $C_{\underline{v}} := \{\underline{v} + \underline{c} \mid \underline{c} \in C\}$ であり, そのコセット代表元はその重み $\mathrm{w}(\underline{w})$ が $C_{\underline{v}}$ の中で最小となる元 $\underline{w} \in C_{\underline{v}}$ である.

任意のコセット代表元 $\underline{v} \notin C$ に対して, $\underline{v}$ を追加した符号 C' は $C' := C \cup C_{\underline{v}}$ と定義される. この C から C' を生成する手続きはコセット追加と呼ばれる. この手続きは重み $\mathrm{w}(\underline{v}) \geq \delta$ をもつコセット代表元 $\underline{v} \notin C'$ がなくなるまで適用可能である. s 回コセット追加を行えば, $[n,k,\delta]_{B,\Pi}$ 符号 C は $[n,k+s,\delta]_{B,\Pi}$ となることが以下の補題により示される.

補題 1 $\underline{v} \notin C$ を $C_{\underline{v}}$ のコセット代表元とする. $\mathrm{w}(\underline{v}) \geq \delta$ ならば, コセット追加符号 $C' := C \cup C_{\underline{v}}$ は $[n,k+1,\delta]_{B,\Pi}$ 符号である.

和田山等 [5] は与えられた検査行列 H によって (通常のランダム誤り訂正符号として) 定義される線形符号 C のすべてのコセットの代表元を見つけるためにシンドロームトレリスを用いている. そこで, コセット代表元を計算するために, シンドロームトレリスと 3 節で述べた複合重み計算トレリスの直積である 3 次元トレリスを考える. これはシンドロームトレリスに任意の $\underline{v} \in F_2^n$ に対して, $\mathrm{w}(\underline{v})$ を計算するためのトレリスを組合わせることにより得られる.

このトレリスには, 各時刻 t, $1 \leq t \leq n$ において状態 $S_{\underline{\sigma}ij}$ を表す $M \times 2^{n-k}$ 個の頂点 $V^t_{\underline{\sigma}ij}$, $\underline{\sigma} \in F_2^{n-k}$, $1 \leq j \leq b_i$, $0 \leq i \leq m$ が存在する. また, 各 t, $1 \leq t \leq n$ に対して, 状態 $S^t_{\underline{\sigma}00}$ と $S^t_{\underline{\sigma}ij}$, $\underline{\sigma} \in F_2^{n-k}$, $1 \leq j \leq b_i$, $1 \leq i \leq m$ はバースト誤りなし, b_i-バーストの j-番目の位置を意味している. これらの頂点 $V^t_{\underline{\sigma}ij}$ は, シンドローム $\underline{\sigma} = (\underline{v}'|0^{n-t})H^T$, $\underline{v}' \in F_2^t$ をもち, t までの部分列が $\underline{v}'$ に一致する任意のパス $\underline{v} \in F_2^n$ に含まれる. ここで, 0^l は長さ l の 0 のベクトルを表すものとし, 演算子 $|$ は 2 つの系列の連結を表している. このトレリスは時刻 $t-1$ の頂点から時刻 t の頂点への辺をもち, その重みは以下の通りである.

(i) $j = b_i$, $0 \leq i \leq m$
  (a) $(V^{t-1}_{\underline{\sigma}ij}, V^t_{\underline{\sigma}00})$, 重み 0,
  (b) $(V^{t-1}_{\underline{\sigma}ij}, V^t_{\underline{\sigma}+\underline{h}_t,k,1})$, 重み $\frac{\pi_i}{b_i}$, $1 \leq k \leq m$,

(ii) $1 \leq j < b_i$, $1 \leq i \leq m$
  (a) $(V^{t-1}_{\underline{\sigma}ij}, V^t_{\underline{\sigma},i,j+1})$, 重み $\frac{\pi_i}{b_i}$,
  (b) $(V^{t-1}_{\underline{\sigma}ij}, V^t_{\underline{\sigma}+\underline{h}_t,i,j+1})$, 重み $\frac{\pi_i}{b_i}$.

さらに, 3 節と同様に計算を容易にするために, 時刻 $n + b_m$ までトレリスを拡張する. ここで, $n < t \leq n + b_m$ に対して対応する辺は以下の通りとなる:

(i) $(V^{t-1}_{\underline{\sigma}ij}, V^t_{\underline{\sigma}00})$, 重み 0, $j = b_i$, $0 \leq i \leq m$
(ii) $(V^{t-1}_{\underline{\sigma}ij}, V^t_{\underline{\sigma},i,j+1})$, 重み $\frac{\pi_i}{b_i}$, $1 \leq j < b_i$, $1 \leq i \leq m$.

以下のアルゴリズムは, 3 次元トレリスにおいて頂点 $V^0_{\underline{0}00}$ から $V^t_{\underline{\sigma}ij}$ への取り得る任意のパス $P(t,\underline{\sigma},i,j)$ に含まれる辺の重みの和の中の最小値である $w(t,\underline{\sigma},i,j)$, $\underline{\sigma} \in F_2^{n-k}$ を t に関して再帰的に計算する.

結果として, コセット代表元 $\mathrm{trunc}_n(P(n+b_m, \underline{\sigma}, 0, 0))$ とその重み $w(n+b_m, \underline{\sigma}, 0, 0)$, $\underline{\sigma} \in F_2^{n-k}$ が計算される. ここで, $\mathrm{trunc}_n(P)$ は系列 P の長さ n までを切り取った部分系列を表している.

アルゴリズム 2 (コセット代表元探索)

Input $H = [\underline{h}_1, \cdots, \underline{h}_n]$.

Output $\mathrm{trunc}_n(P(n+b_m, \underline{\sigma}, 0, 0))$, $\underline{\sigma} \in F_2^{n-k}$, $w(n+b_m, \underline{\sigma}, 0, 0)$, $\underline{\sigma} \in F_2^{n-k}$.

Step 1 Set $w(0, \underline{0}, 0, 0) := 0$, $w(t, \underline{\sigma}, i, j) := \infty$ for $0 \leq t \leq n + b_m$, $\underline{\sigma} \in F_2^{n-k}$,

$1 \le j \le b_i,\ 1 \le i \le m;$
$P(0,\underline{\sigma},0,0) := \emptyset$ for $\underline{\sigma} \in F_2^{n-k}$; $t := 1$;

Step 2 Calculate

$$w(t,\underline{\sigma},i,j) \qquad (1)$$

$$:= \begin{cases} 1 : (i=0, j=0) \\ \min\limits_{0\le k\le m}\{w(t-1,\underline{\sigma},k,b_k)\}; \\ 2 : (i\neq 0, j=1) \\ \min\limits_{0\le k\le m}\{w(t-1,\underline{\sigma}-\underline{h}_t,k,b_k)\}+\pi_i/b_i; \\ 3 : (i\neq 0, j\neq 1) \\ \min\limits_{x\in\{0,1\}}\{w(t-1,\underline{\sigma}-x\cdot\underline{h}_t,i,j-1)\} \\ \qquad +\pi_i/b_i; \end{cases}$$

$$i' := \arg\min_{0\le k\le m}\{w(t-1,\underline{\sigma},k,b_k)\};$$
$$i'' := \arg\min_{0\le k\le m}\{w(t-1,\underline{\sigma}-\underline{h}_t,k,b_k)\};$$
$$w_0 := w(t-1,\underline{\sigma},i,j-1);$$
$$w_1 := w(t-1,\underline{\sigma}-\underline{h}_t,i,j-1);$$

$$P(t,\underline{\sigma},i,j) \qquad (2)$$

$$:= \begin{cases} 1 : (i=0, j=0) \\ P(t-1,\underline{\sigma},i',b_{i'})|0; \\ 2 : (i\neq 0, j=1) \\ P(t-1,\underline{\sigma}-\underline{h}_t,i'',b_{i''})|1; \\ 3 : (i\neq 0, j\neq 1) \\ P(t-1,\underline{\sigma},i,j-1)|0; \quad \text{if}(w_0 \le w_1), \\ P(t-1,\underline{\sigma}-\underline{h}_t,i,j-1)|1; \quad \text{if}(w_0 > w_1). \end{cases}$$

Step 3 If $t < n+b_m$, then $t := t+1$ and go to Step 2.

Step 4 Output a coset leader $\text{trunc}_n(P(n+b_m,\underline{\sigma},0,0))$ and $w(n+b_m,\underline{\sigma},0,0)$, $\underline{\sigma} \in F_2^{n-k}$.

一般に最小重みをもつパスは複数存在するが, アルゴリズムにおいては単純に最初にみつけたパスを選択することにする.

このアルゴリズムの計算量は $O(2^{n-k}nmM)$ となる. アルゴリズム 2 を用いて, 複合誤り訂正符号を生成するためのコセット追加アルゴリズムを与えることができる.

アルゴリズム 3 (コセット追加)

Input A generator matrix $G^{(0)}$ of the code C.

Output A generator matrix $G^{(i)}$ of the new code $C^{(i)}$.

Step 1 Set $i := 0$.

Step 2 Calculate the check matrix $H^{(i)}$ from $G^{(i)}$.

Step 3 Find a heaviest coset leader $\underline{v}^*$ by applying Algorithm 2 for $H^{(i)}$.

Step 4 If $\text{w}(\underline{v}^*) \ge \delta$ then $i := i+1$,

$$G^{(i)} := \begin{bmatrix} G^{(i-1)} \\ \underline{v}^* \end{bmatrix}$$

and go to Step 3 else stop.

次にアルゴリズム 2 を修正し, コセット重み計算と同時に符号 C の最小距離を求める方法を提案する. シンドローム $\underline{\sigma} = \underline{0}$ をもつコセットはちょうど符号 C となる. $w(t,\underline{0},0,0) = 0$, $0 \le t \le n+b_m$ は $\underline{v} = \underline{0} \in C$ に対応する.

一方, 各時刻 t において非零語 $\underline{v} \in F_2^t \setminus \{0^t\}$ に対して, $\text{trunc}_n(\underline{v}|0^{n+b_m-t})H^T = \underline{0}$ を満たし, 状態 $S_{\underline{0}kb_k}$, $0 \le k \le m$ を通る任意のパスの重みの最小値 $\text{mw}(t)$ は $\text{mw}(t-1)$ あるいは $\min_{1\le k\le m}\{w(t,\underline{0},k,b_k)\}$, $0 \le t \le n+b_m$, として得られる.

ここで前の値 $\text{mw}(t-1)$ は $\text{trunc}_n(\underline{w}|0^{n+b_m-t+1})H^T = \underline{0}$ を満たし, 重み $\text{mw}(t-1)$ をもつ非零語 $\underline{w} \in F_2^{t-1} \setminus \{0^{t-1}\}$ に対応し, 全零のパス 0^t のみしか存在しない場合には $\text{mw}(t) := \infty$ である.

このように, アルゴリズム 2 と同じ計算量 $O(2^{n-k}nmM)$ でコセット代表元の計算と同時に最小距離 $\delta = mw(n+b_m)$ を計算する以下のアルゴリズムを与えることが出来る.

アルゴリズム 4 (最小距離計算)

Input $H = [\underline{h}_1, \cdots, \underline{h}_n]$.

Output The minimum distance $\delta = \text{mw}(n+b_m)$.

Step 1 Set $w(0,\underline{0},0,0) := 0$, $w(t,\underline{\sigma},i,j) := \infty$ for $1 \le t \le n+b_m$, $\underline{\sigma} \in F_2^{n-k}$,

$1 \le j \le b_i, 1 \le i \le m;$
$P(0,\underline{\sigma},0,0) := \emptyset$ for $\underline{\sigma} \in F_2^{n-k}$;
$\mathrm{mw}(0) := \infty; \quad t := 1;$

Step 2 Calculate (1), (2) as in Step 2 of Algorithm 2 and

$$\mathrm{mw}(t) := \min\{\mathrm{mw}(t-1), \min_{1\le k\le m}\{w(t,\underline{0},k,b_k)\}\}.$$

Step 3 If $t < n + b_m$, then $t := t + 1$ and go to Step 2.

Step 4 Output $\mathrm{mw}(n + b_m)$.

コセット追加の表記を少し一般化する. まず, 与えられた複合誤り訂正符号 C の最小距離 δ 以下の定数 δ' を決める. このとき, $w(\underline{v}) \ge \delta'$ を満たす任意のコセット代表元 $\underline{v} \notin C$ に対し, 符号 $C' := C \cup C_{\underline{v}}$ を得ることが出来る. ここで, $C_{\underline{v}}$ は $\underline{v}$ を含むコセットである. 明らかに符号 C' の最小距離 $\ge \delta'$ である. このような構成を設計距離 δ' によるコセット追加と呼ぶ. これにより, 複合誤り訂正能力を多少犠牲にするが, より大きな符号化率を得ることができ, よい符号が見つかる可能性が高まると考えられる.

例 5 コセット追加により得られたいくつかの複合誤り訂正符号を表 2 に示す. 元の符号として符号長 $n = 27$ の巡回符号を用いている. 表中の δ' は設計距離であり, k' は新しく得られた符号の次元である. 表 2 のコセット追加符号と他の巡回符号を比較した

表 2: 巡回符号とそのコセット追加符号 (符号長 $n = 27$)

Original cyclic codes			Burst lengths, weights		Augmented codes	
k	d	δ	B	Π	k'	δ'
9	10	7.7	$\{1,2,6\}$	$\{1.0,1.9,2.9\}$	12	5.9
9	10	6.8	$\{1,2,6\}$	$\{1.0,1.4,3.0\}$	11	6.2
9	10	8.5	$\{1,3,5\}$	$\{1.0,2.7,3.5\}$	10	7.5

結果を表 3 にまとめる. 符号長 $n = 27$ とバースト長集合 B が同じ場合 (バースト重み集合 Π は異なる場合がある) について, これらの符号の訂正可能な複合誤りパターンを示した. 結果として, 表 3 の符号は符号率, あるいは, 複合誤り訂正能力において優れていることが分かる.

表 3: コセット追加符号とその性能 (符号長 $n = 27$)

B	Augmented codes		Other cyclic codes	
	k	error patterns	k	error patterns
$\{1,2,6\}$	12	$1\times 1b + 1\times 2b$, $1\times 6b$	12	$2\times 1b$
$\{1,2,6\}$	11	$3\times 1b, 2\times 2b$, $1\times 6b$	11	$1\times 1b + 1\times 2b$, $1\times 6b$
$\{1,3,5\}$	10	$3\times 1b, 1\times 1b +$ $1\times 3b, 1\times 5b$	10	$3\times 1b, 1\times 5b$

5 復号法

任意の複合誤り訂正符号 C に対する複合アルゴリズムを提案する. ここで, 符号 C は検査行列 $H = [\underline{h}_1, \cdots, \underline{h}_n]$ により定義され, 最小距離は δ であるとする. さらに, $\underline{v} \in F_2^n$ を受信語とする.

コセット追加の場合と同様に, 各時刻 t, $1 \le t \le n$ に対して, $M \times 2^{n-k}$ 個の状態 $S_{\underline{\sigma}ij}$, $\underline{\sigma} \in F_2^{n-k}$, $1 \le j \le b_i$, $0 \le i \le m$ に対応する頂点 $V^t_{\underline{\sigma}ij}$ をもつ 3 次元トレリスを考える. このトレリスは以下に定義される時刻 $t-1$ の頂点から時刻 t の頂点への辺とその重みをもつ.

(i) $j = b_i$, $0 \le i \le m$
  (a) $(V^{t-1}_{\underline{\sigma}ij}, V^t_{\underline{\sigma}00})$, 重み 0,
    $(V^{t-1}_{\underline{\sigma}ij}, V^t_{\underline{\sigma}+\underline{h}_t,k,1})$, 重み $\frac{\pi_k}{b_k}$, $1 \le k \le m$,
  (b) $(V^{t-1}_{\underline{\sigma}ij}, V^t_{\underline{\sigma}+\underline{h}_t,0,0})$, 重み 0,
    $(V^{t-1}_{\underline{\sigma}ij}, V^t_{\underline{\sigma}k1})$, 重み $\frac{\pi_k}{b_k}$, $1 \le k \le m$,
(ii) $1 \le j < b_i$, $1 \le i \le m$
    $(V^{t-1}_{\underline{\sigma}ij}, V^t_{\underline{\sigma},i,j+1})$, 重み $\frac{\pi_i}{b_i}$,
    $(V^{t-1}_{\underline{\sigma}ij}, V^t_{\underline{\sigma}+\underline{h}_t,i,j+1})$, 重み $\frac{\pi_i}{b_i}$.

さらに, 3 節と同様に計算を容易にするため, 時刻 $n + b_m$ までトレリスを拡張する. ここで, $n < t \le n + b_m$ に対する辺は以下の通りである:

(i) $(V^{t-1}_{\underline{\sigma}ij}, V^t_{\underline{\sigma}00})$, 重み 0, $j = b_i$, $0 \le i \le m$,
(ii) $(V^{t-1}_{\underline{\sigma}ij}, V^t_{\underline{\sigma},i,j+1})$, 重み $\frac{\pi_i}{b_i}$, $1 \le j < b_i$, $1 \le i \le m$.

以下のアルゴリズムは, 3 次元トレリスにおいて頂点 $V^0_{\underline{0}00}$ から $V^t_{\underline{\sigma}ij}$ への任意のパス $P(t,\underline{\sigma},i,j)$ に含まれる辺の重みの和の中の最小値である $w(t,\underline{\sigma},i,j)$,

$\underline{\sigma} \in F_2^{n-k}$ を t に関して再帰的に計算する．最終的に，受信語から最小距離にある符号語と 0^{b_m} を連結した系列である $P(n+b_m, \underline{0}, 0, 0)$ を得る．

$$\rho(x_1, x_2) = \begin{cases} 0 & \text{if} \quad x_1 = x_2 \\ 1 & \text{if} \quad x_1 \neq x_2 \end{cases}.$$

とする．

アルゴリズム 5 (復号)

Input Check matrix: $H = [\underline{h}_1, \cdots, \underline{h}_n]$;
A received word: $\underline{v} = (v_i)_{1 \le i \le n}$;

Output $\hat{\underline{c}} = \mathrm{trunc}_n(P(n+b_m, \underline{0}, 0, 0))$;

Step 1 Initialization:
$\delta(0, \underline{0}, 0, 0) := 0$, $\delta(t, \underline{\sigma}, i, j) := \infty$
for $0 \le t \le n+b_m$, $\underline{\sigma} \in F_2^{n-k}$,
$1 \le j \le b_i$, $1 \le i \le m$;
$P(0, \underline{\sigma}, 0, 0) := \emptyset$ for $\underline{\sigma} \in F_2^{n-k}$;
$t := 1$;

Step 2 For $\underline{\sigma} \in F_2^n$, $1 \le j \le b_i$, $0 \le i \le m$, calculate the following values.

$$\delta(t, \underline{\sigma}, i, j) := \begin{cases} 1: \ (i=0, j=0) \\ \quad \min\limits_{0 \le k \le m} \{\delta(t-1, \underline{\sigma} - \rho(0, v_t) \cdot \underline{h}_t, k, b_k)\}; \\ 2: \ (i \neq 0, j=1) \\ \quad \min\limits_{0 \le k \le m} \{\delta(t-1, \underline{\sigma} - \rho(1, v_t) \cdot \underline{h}_t, k, b_k)\} \\ \qquad\qquad + \pi_i / b_i; \\ 3: \ (i \neq 0, j \neq 1) \\ \quad \min\limits_{x \in \{0,1\}} \{\delta(t-1, \underline{\sigma} - x \cdot \underline{h}_t, i, j-1)\} \\ \qquad\qquad + \pi_i / b_i; \end{cases}$$

$$i' := \arg \min_{0 \le k \le m} \{\delta(t-1, \underline{\sigma} - \rho(0, v_t) \cdot \underline{h}_t, k, b_k)\};$$

$$i'' := \arg \min_{0 \le k \le m} \{\delta(t-1, \underline{\sigma} - \rho(1, v_t) \cdot \underline{h}_t, k, b_k)\};$$

$$\delta_0 := \delta(t-1, \underline{\sigma}, i, j-1);$$

$$\delta_1 := \delta(t-1, \underline{\sigma} - \underline{h}_t, i, j-1);$$

$$P(t, \underline{\sigma}, i, j) := \begin{cases} 1: (i=0, j=0) \\ \quad P(t-1, \underline{\sigma} - \rho(0, v_t) \cdot \underline{h}_t, i', b_{i'}) | 0; \\ 2: (i \neq 0, j=1) \\ \quad P(t-1, \underline{\sigma} - \rho(1, v_t) \cdot \underline{h}_t, i'', b_{i''}) | 1; \\ 3: (i \neq 0, j \neq 1) \\ \quad P(t-1, \underline{\sigma}, i, j-1) | 0; & \text{if}(\delta_0 \le \delta_1), \\ \quad P(t-1, \underline{\sigma} - \underline{h}_t, i, j-1) | 1; & \text{if}(\delta_0 > \delta_1). \end{cases}$$

Step 3 If $t < n + b_m$, then $t := t+1$ and go to Step 2.

Step 4 Output the estimated codeword $\mathrm{trunc}_n(P(n+b_m, \underline{0}, 0, 0))$.

各時刻 t, $1 \le t \le n+b_m$ において，$M \times 2^{n-k}$ 個の頂点に対して m 回の比較と加算を行うので，上記の復号アルゴリズムの計算量は $O(2^{n-k} nmM)$ となる．

例 6 バースト長と重みを $B = \{1, 2, 6\}$, $\Pi = \{1.0, 1.4, 3.0\}$ として得られる $[27, 11, 8, 6.2]_{B,\Pi}$ 符号を考える．これは，[27, 9, 10] 準巡回符号からコセット追加により構成した符号 [6] であり，訂正可能な誤りパターンは，$(3 \times 1b, 2 \times 2b, 1 \times 6b)$ である．すなわち，3 個のランダム誤り，2 個の長さ 2 バースト誤り，1 個の長さ 6 バースト誤りが訂正可能である．これは，通常のハミング距離に基づいた最小距離復号では，3 ランダム誤り訂正符号であり，バースト誤り訂正符号としては長さ 6 のバースト誤りを 1 つ訂正可能である．上記のアルゴリズムを用いて，ランダム誤り訂正符号，バースト誤り訂正符号，および，複合誤り訂正符号とみなして復号を行い，複合誤り訂正の特徴を調べる．通信路として，下記の *Gilbert-Elliot* 通信路を仮定する．

p	状態 G から状態 B への遷移確率
q	状態 B から状態 G への遷移確率
P_B	状態 G における誤り発生率
P_G	状態 B における誤り発生率

表 4 に示された条件の下での復号後のビット誤り率のシミュレーション結果を表 5 に示す．どの条件に対しても，ランダム誤り訂正に比べ複合誤り訂正のビット誤り率は 1/3 程度になっており，その有効性が示されている．

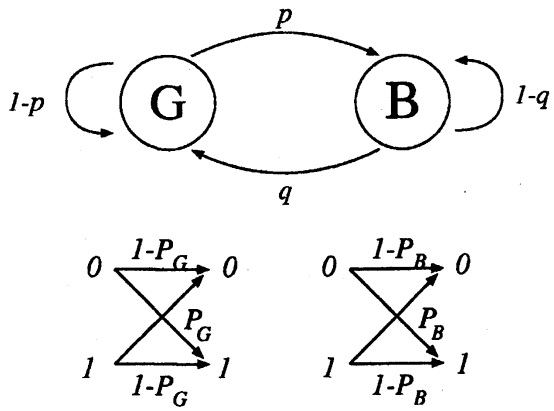

図 3: Gilbert-Elliot channel

表 4: シミュレーション条件

Type	p	q	P_G	P_B	BER
A	0.0080	0.65	0.0001	0.90	0.011041
B	0.0080	0.55	0.0001	0.80	0.011569
C	0.0105	0.72	0.0003	0.75	0.011076

6 まとめ

複合誤り訂正符号の性能の調査に関連して, 4 つのアルゴリズムを提案した. まず, 任意の語に対して複合重みを計算する方法を示し, これを拡張することにより, 与えられた複合誤り訂正符号に対して同じ最初距離をもち, より高い符号化率をもつ良い符号を得るためのコセット追加アルゴリズムを与えた. さらに, コセット代表元とその重みの計算と同時に最小距離を計算する方法を与えた. 最後にコセット追加アルゴリズムを修正し, 任意の複合誤り訂正符号の復号法を提案した. これらのアルゴリズムの適用の産物として, いくつかの良い複合誤り訂正符号を示した.

表 5: 復号シミュレーション結果

	A	B	C
Random	0.000500	0.001056	0.000241
Burst	0.000519	0.000370	0.000315
Compound	0.000167	0.000241	0.000074

参考文献

[1] E.M. Gabidulin, "Combinational metrics in coding theory", *2nd Intern. Symp. Information Theory*, Armenia, USSR, 1971 (Eds. B.N. Petrov and F. Csaki), Hungarian Academy of Science, pp.169–175, 1972.

[2] Mitsuru Hamada, "A note on combinatorial metrics for error-correcting codes", Technical Report of IEICE, IT99-31, pp.97–102, 1997.

[3] 栗原正純, "複合誤り制御に関する研究", 電気通信大学博士論文, 2002.

[4] Shojiro Sakata, "Step-by-step decoding of compound-error-correcting codes", presented at the 2001 Intern. IEEE Symposium Information Theory, Washington D.C., June 2001, and submitted for publication in IEEE Trans. Inform. Theory.

[5] Tadashi Wadayama, Hiroyuki Kadokawa, "An algorithm for augmenting a binary linear code up to Gilbert bound and new codes obtained by the algorithm", IEICE Trans. Fundamentals, Vol.E85-A, pp.2196–2202, Oct. 2002.

[6] Masaya Fujisawa, Shusuke Maeda, Shojiro Sakata, "Compound-error-correcting codes and their augmentation", IEICE Trans. Fundamentals, Vol.E86-A, pp.1813–1819, July 2003.

[7] Tadao Kasami, "Optimum shortened cyclic codes for burst-error correction", IEEE Trans. Inform. Theory, pp.105–109, 1963.

[8] Henk van Tilborg, "On quasi-cyclic codes with rate 1/m", IEEE Trans. Inform. Theory, vol.24, pp.628–630, Sept. 1978.